facebook

Artificial Intelligence Research

Video Understanding

Georgia Gkioxari

Content

- Intro

- Video Architectures
  - ConvNets + LSTM
  - 3D ConvNets
  - Two-Stream

- Video Understanding Tasks
  - Action Classification, Action Localization, Tracking

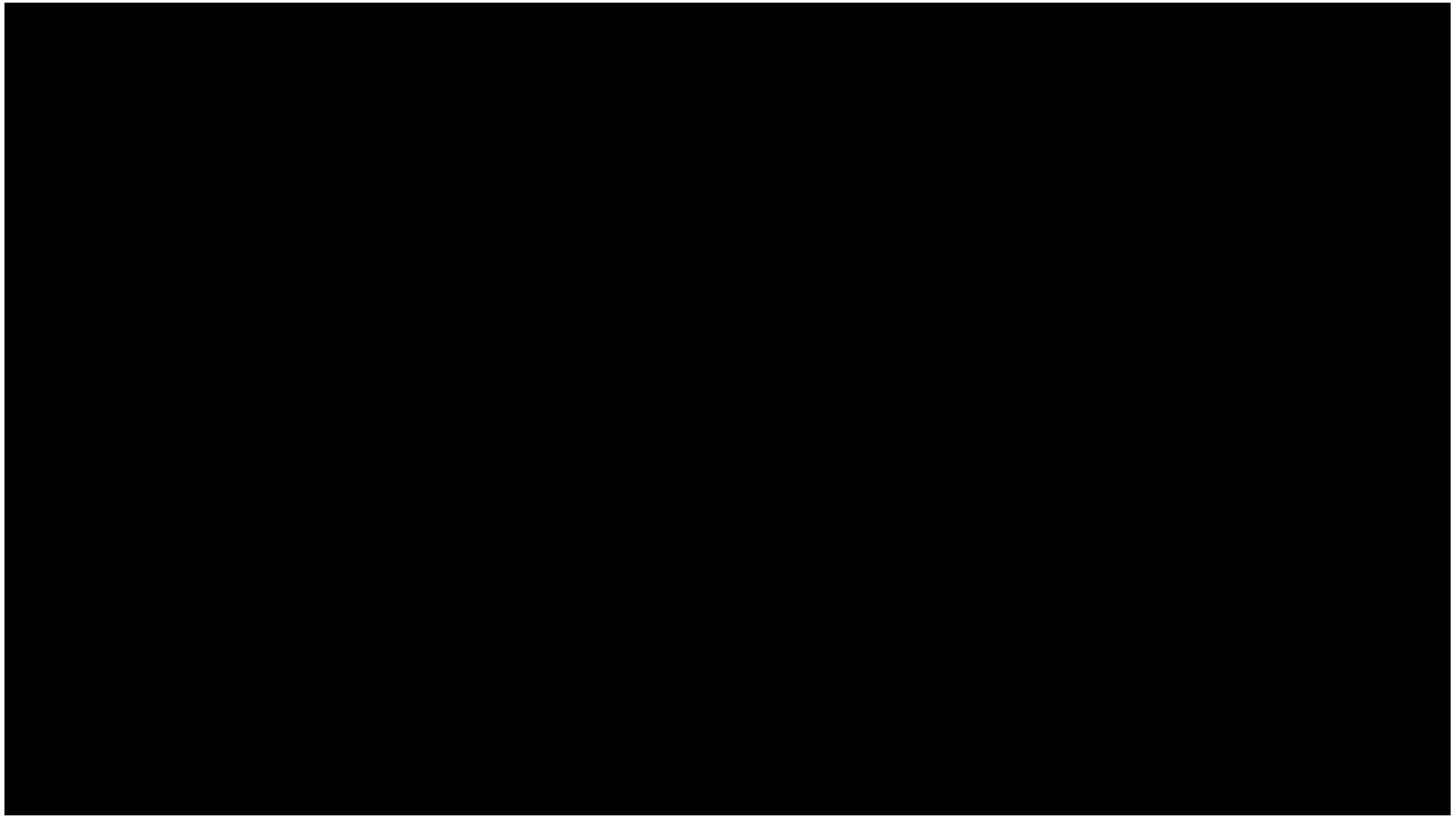
Content

- Intro

- Video Architectures
  - ConvNets + LSTM
  - 3D ConvNets
  - Two-Stream

- Video Understanding Tasks
  - Action Classification, Action Localization, Tracking

Motion



Johansson, Biological Motion Perception

Motion

Heider-Simmel

Video Understanding

- Perception in (x, y, t)
  - Frames (x,y) provide appearance cues
  - Time (t) provides motion cues

- Video understanding is the science of modeling appearance (spatio-) with motion (temporal-) in order to understand actions and intent (reasoning)

Action Classification

Task

Given a video V, predict its action label, e.g. drumming, chopping onion

Metrics

Top-k accuracy

Content

- Intro

- Video Architectures
  - ConvNets + LSTM
  - 3D ConvNets
  - Two-Stream

- Video Understanding Tasks
  - Action Classification, Action Localization, Tracking

Action Classification: Challenges

- Spatio-temporal processing
  - Convolutions perform spatial operations – capture appearance
  - What is the equivalent for temporal operations?
- Motion representations
  - Naively processing all frames in a video is expensive and inefficient
  - How should we represent motion?
- Fusion of appearance and motion cues
  - To recognize actions we need to combine appearance (spatio-) and motion (temporal-) cues.
  - How should this fusion be operationalized?

Video Architectures

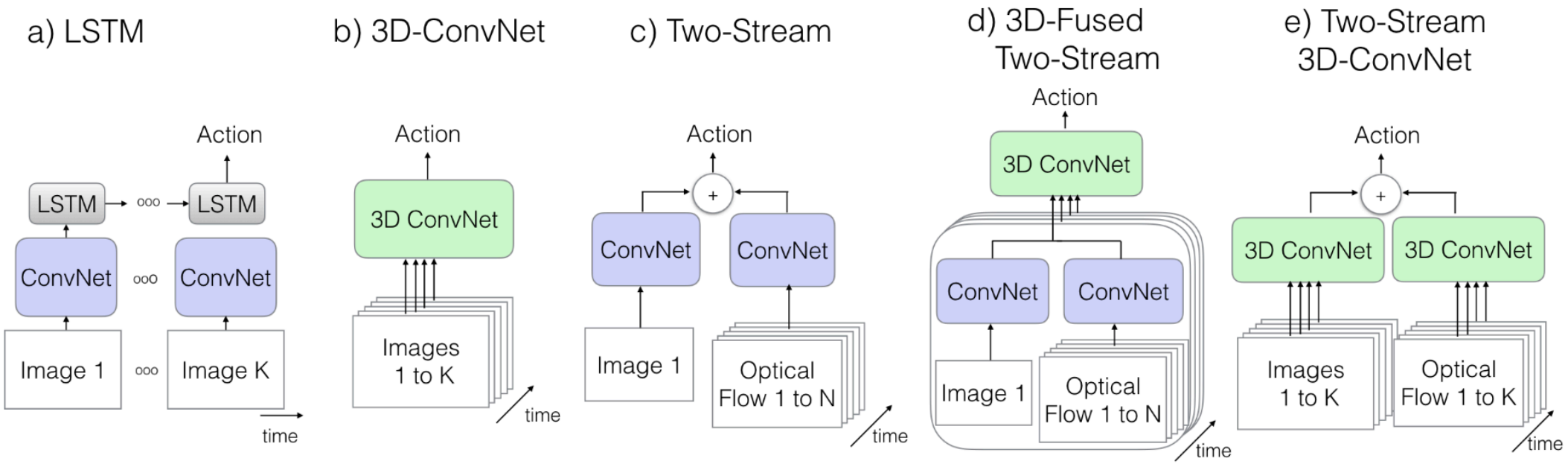


Figure from Carreira & Zisserman, Quo Vadis, Action Recognition? A new model and the Kinetics dataset.

Video Architectures

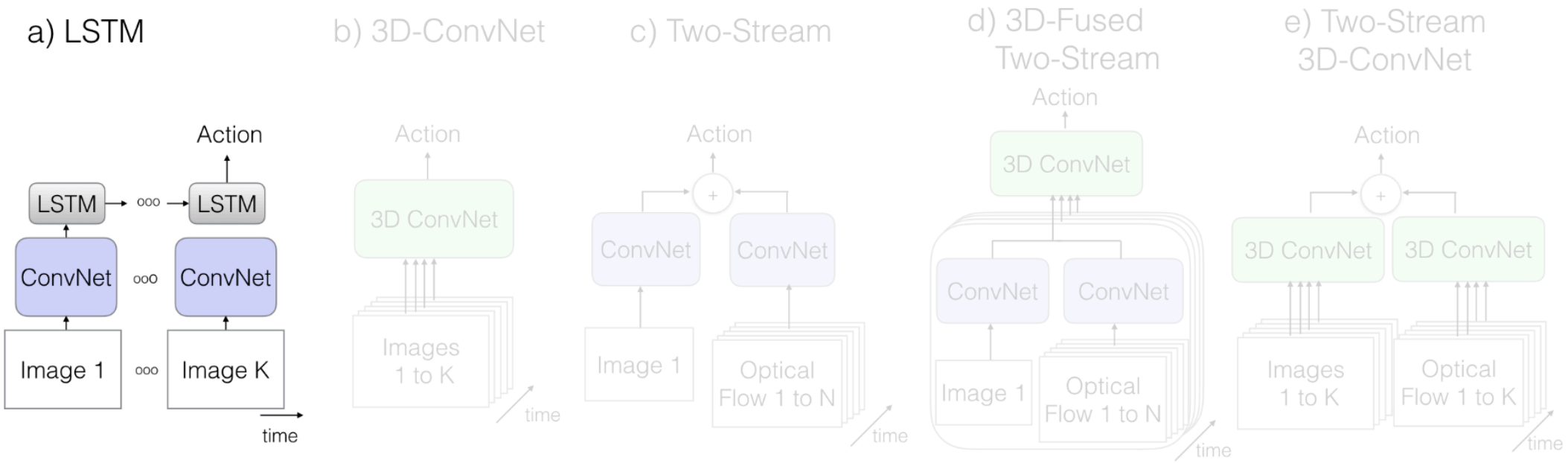


Figure from Carreira & Zisserman, Quo Vadis, Action Recognition? A new model and the Kinetics dataset.

Video Architectures

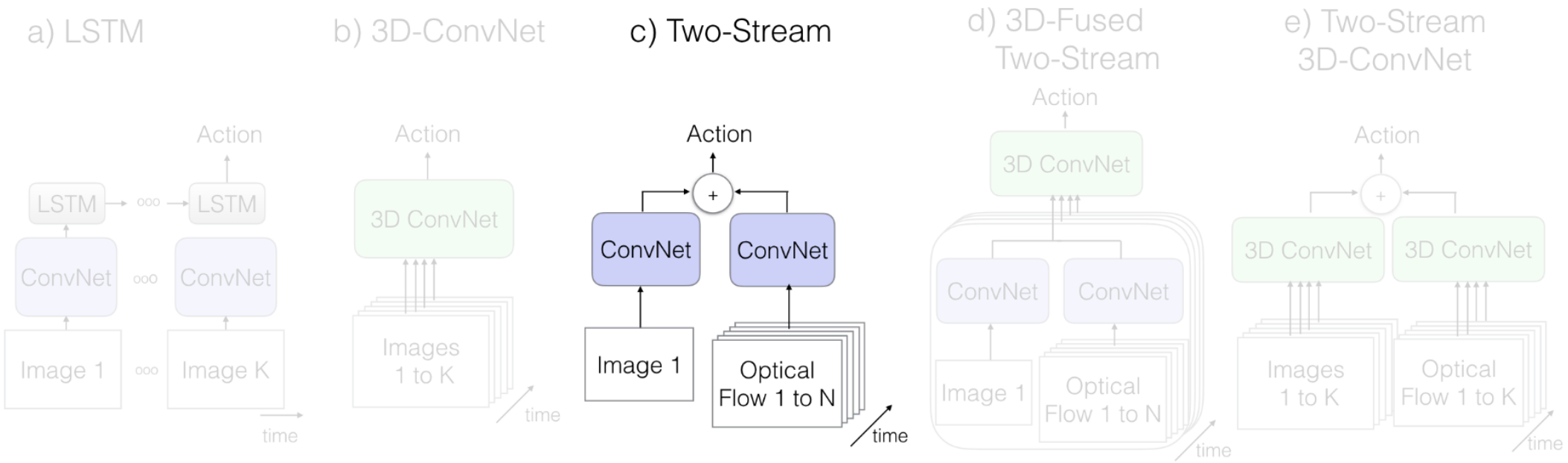
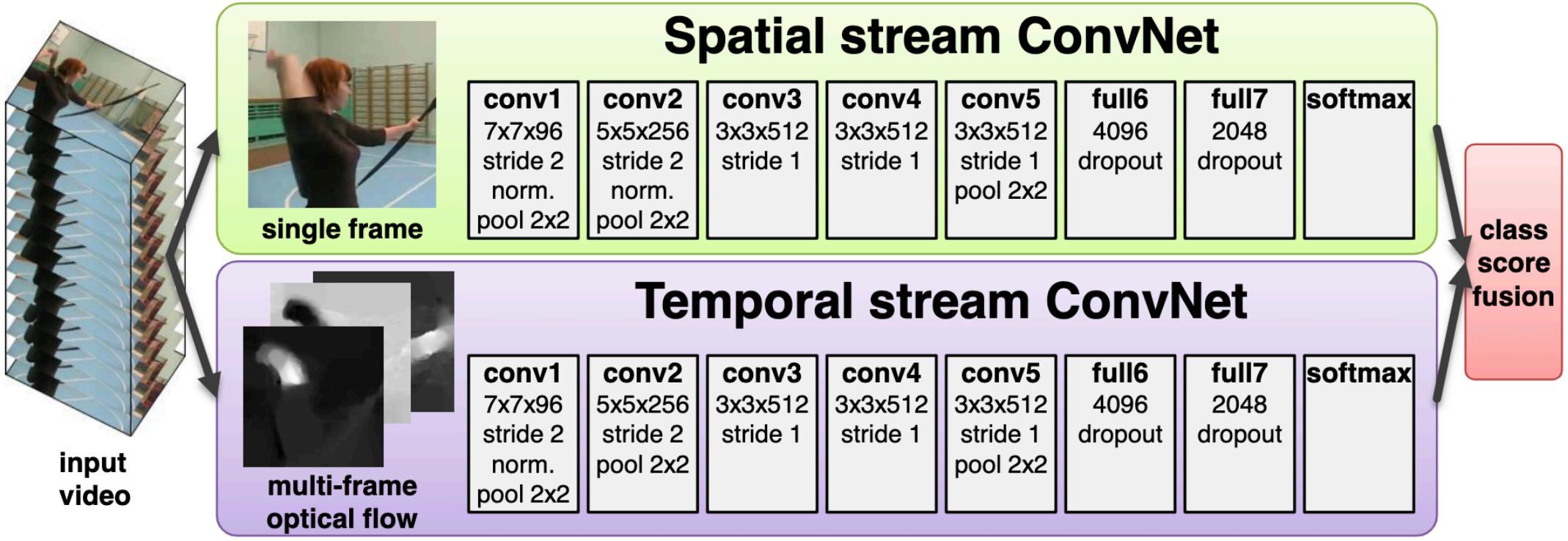


Figure from Carreira & Zisserman, Quo Vadis, Action Recognition? A new model and the Kinetics dataset.

Action Classification

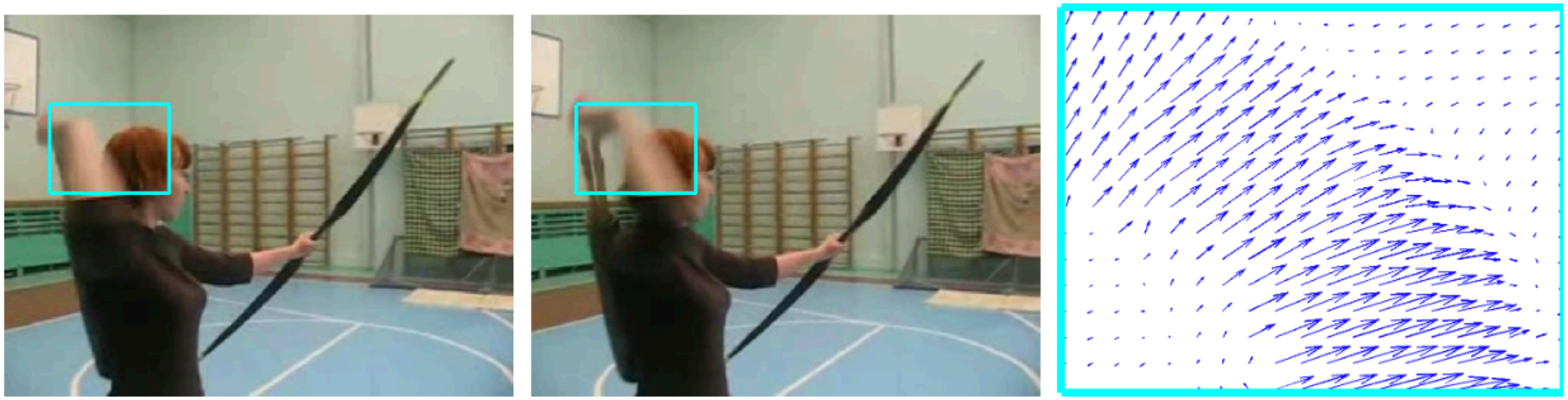
Two-Stream Models



Simonyan & Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos

Action Classification

Optical Flow



Simonyan & Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos

Action Classification

UCF-101

(a) **Spatial ConvNet.**

Training setting	Dropout ratio	
	0.5	0.9
From scratch	42.5%	52.3%
Pre-trained + fine-tuning	70.8%	**72.8%**
Pre-trained + last layer	**72.7%**	59.9%

(b) **Temporal ConvNet.**

Input configuration	Mean subtraction	
	off	on
Single-frame optical flow ($L = 1$)	-	73.9%
Optical flow stacking (1) ($L = 5$)	-	80.4%
Optical flow stacking (1) ($L = 10$)	79.9%	**81.0%**
Trajectory stacking (2)($L = 10$)	79.6%	80.2%
Optical flow stacking (1)($L = 10$), bi-dir.	-	**81.2%**

Simonyan & Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos

Action Classification

UCF-101

Spatial ConvNet	Temporal ConvNet	Fusion Method	Accuracy
Pre-trained + last layer	bi-directional	averaging	85.6%
Pre-trained + last layer	uni-directional	averaging	85.9%
Pre-trained + last layer	uni-directional, multi-task	averaging	86.2%
Pre-trained + last layer	uni-directional, multi-task	SVM	**87.0%**

Simonyan & Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos

Video Architectures

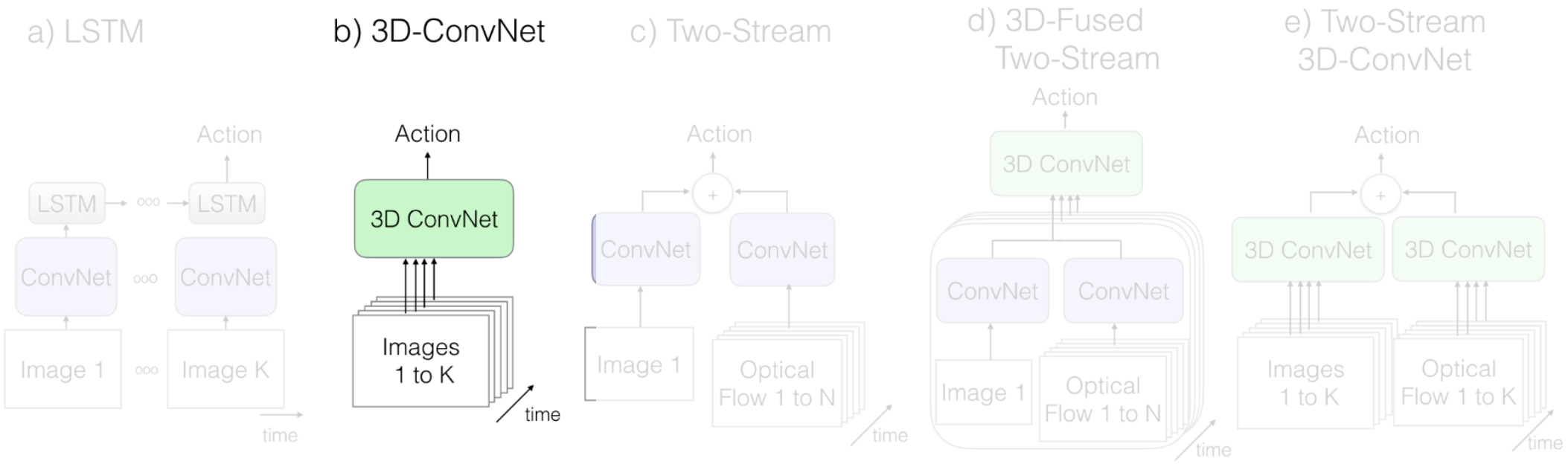


Figure from Carreira & Zisserman, Quo Vadis, Action Recognition? A new model and the Kinetics dataset.

Video Architectures

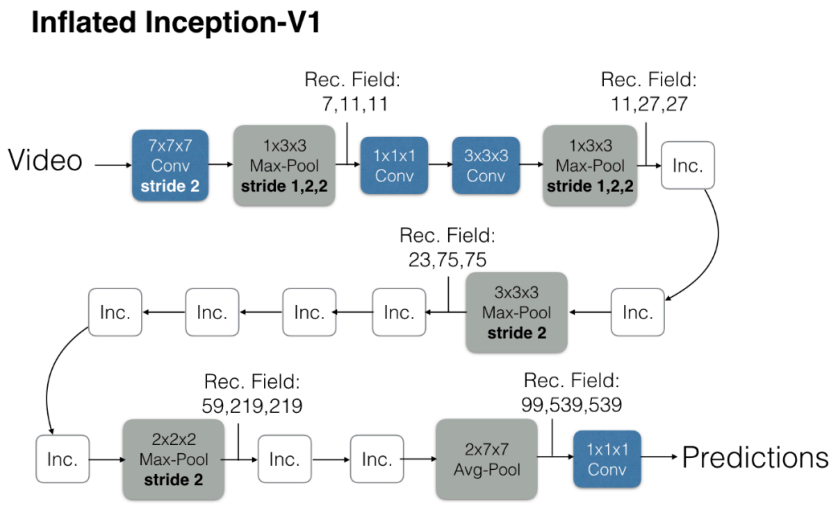
3D ConvNets

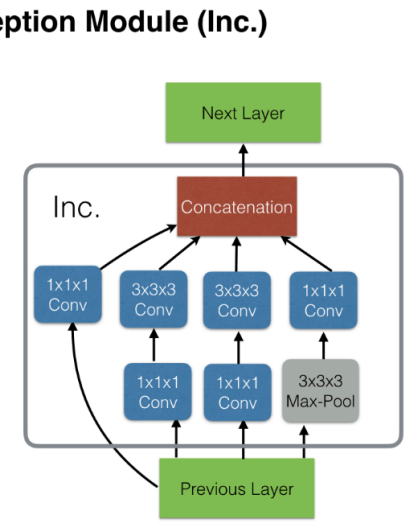
- Input is a 3D tensor of size T x H x W x 3 instead of H x W x 3
- Make all (2D) convolutions 3D, e.g. 3x3 convolutions become 3x3x3
- Pooling and strides also become 3D, e.g. 2x2 maxpool become 2x2x2
- All intermediate feature maps are 3D (of shape t x h x w)

Video Architectures

Inflated 3D ConvNets (I3D)

- The architecture of the 3D ConvNet follows the architecture of a 2D ConvNet by inflating its 2D weights, e.g. Inception 3D





Video Architectures

Inflated 3D ConvNets (I3D)

- The architecture of the 3D ConvNet follows the architecture of a 2D ConvNet by inflating its 2D weights, e.g. Inception 3D
- 3D ConvNets can be initialized with ImageNet weights
  - Copy 2D weights to each time slot in the 3D weights (inflate)
- Thus, 3D ConvNets benefit from the 2D designs and their learned parameters on ImageNet

Video Architectures

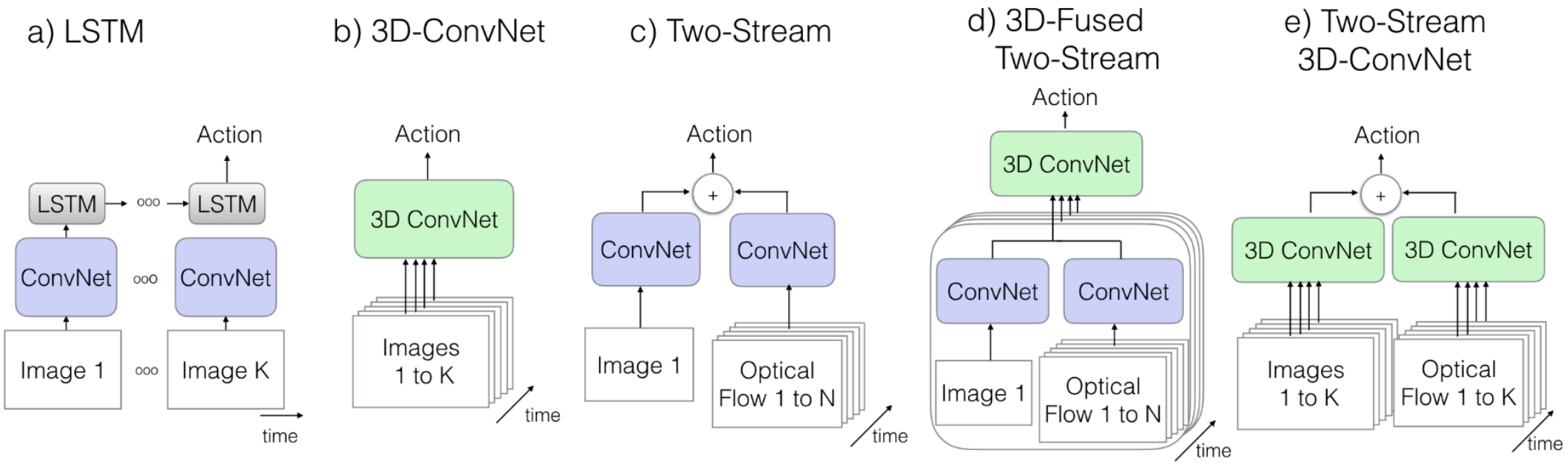
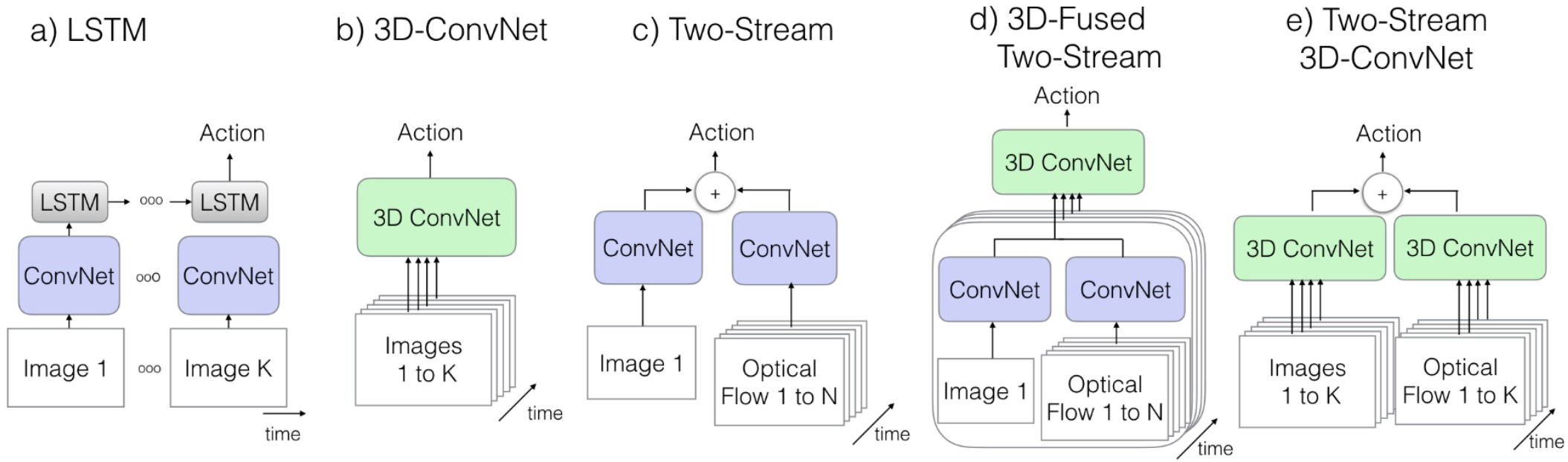


Figure from Carreira & Zisserman, Quo Vadis, Action Recognition? A new model and the Kinetics dataset.



Architecture	UCF-101			HMDB-51			Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	63.3	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	56.1	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	62.2	52.4	65.6
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	–	–	67.2
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**71.1**	**63.4**	**74.2**

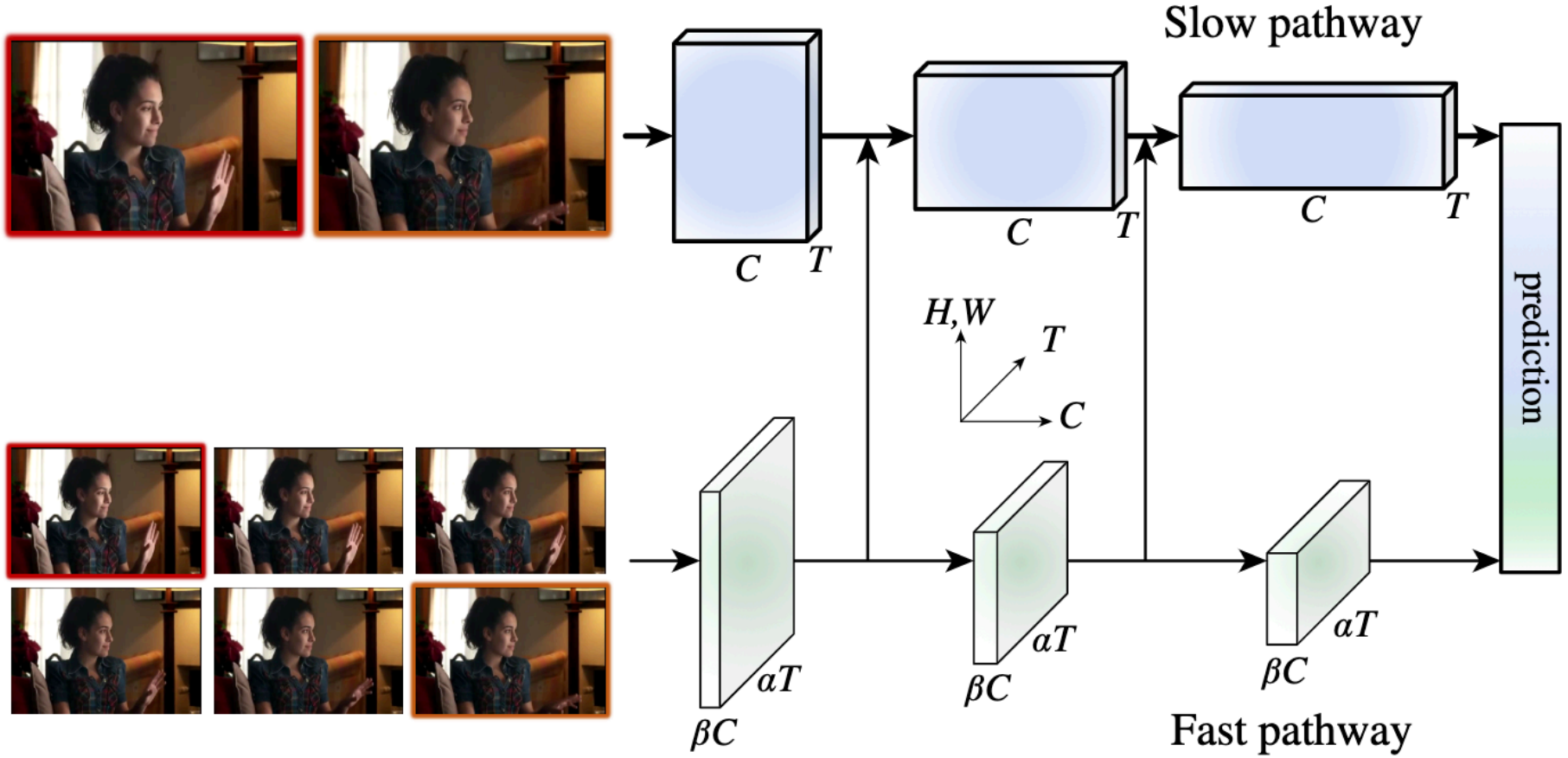
Architecture	Kinetics			ImageNet then Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	53.9	–	–	63.3	–	–
(b) 3D-ConvNet	56.1	–	–	–	–	–
(c) Two-Stream	57.9	49.6	62.8	62.2	52.4	65.6
(d) 3D-Fused	–	–	62.7	–	–	67.2
(e) Two-Stream I3D	**68.4** (88.0)	**61.5** (83.4)	**71.6** (90.0)	**71.1** (89.3)	**63.4** (84.9)	**74.2** (91.3)

Video Architectures

Two-Models and 3D ConvNets are inefficient

- Optical flow computation for every pair of frames is done offline (pre-processing)
- Storing optical flow frames is a bottleneck
- Two-stream models have 2x the number of parameters
- Appearance and motion cues are learned disjointly
- 3D operations (e.g. convolutions) on big 3D tensors are expensive (memory and time)

Slow Fast Networks

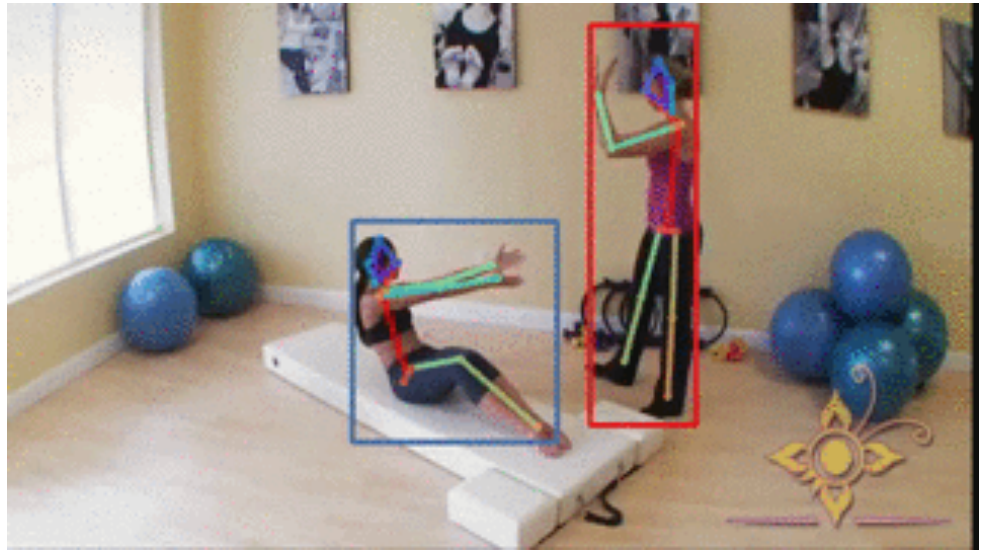


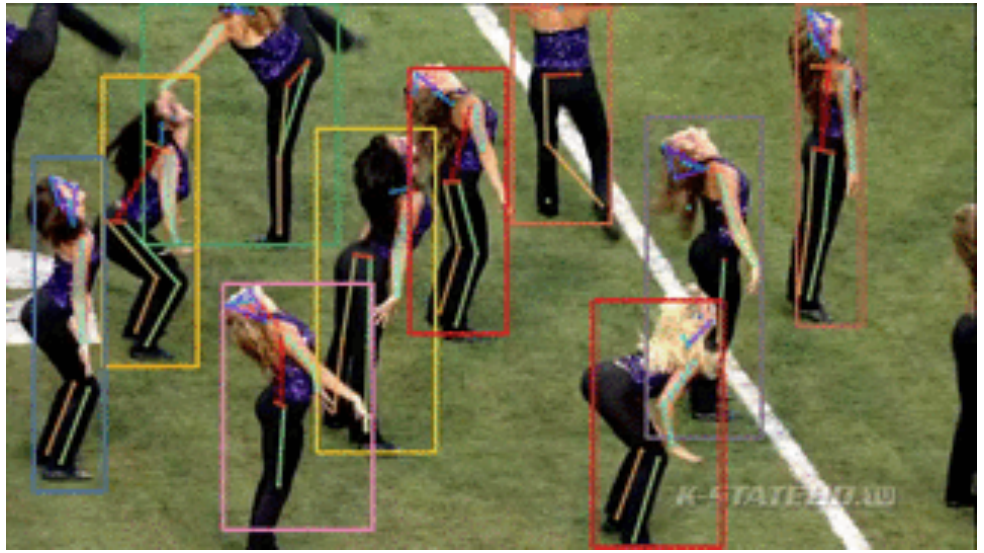
Feichtenhofer et al., Slow Fast Networks for Video Recognition

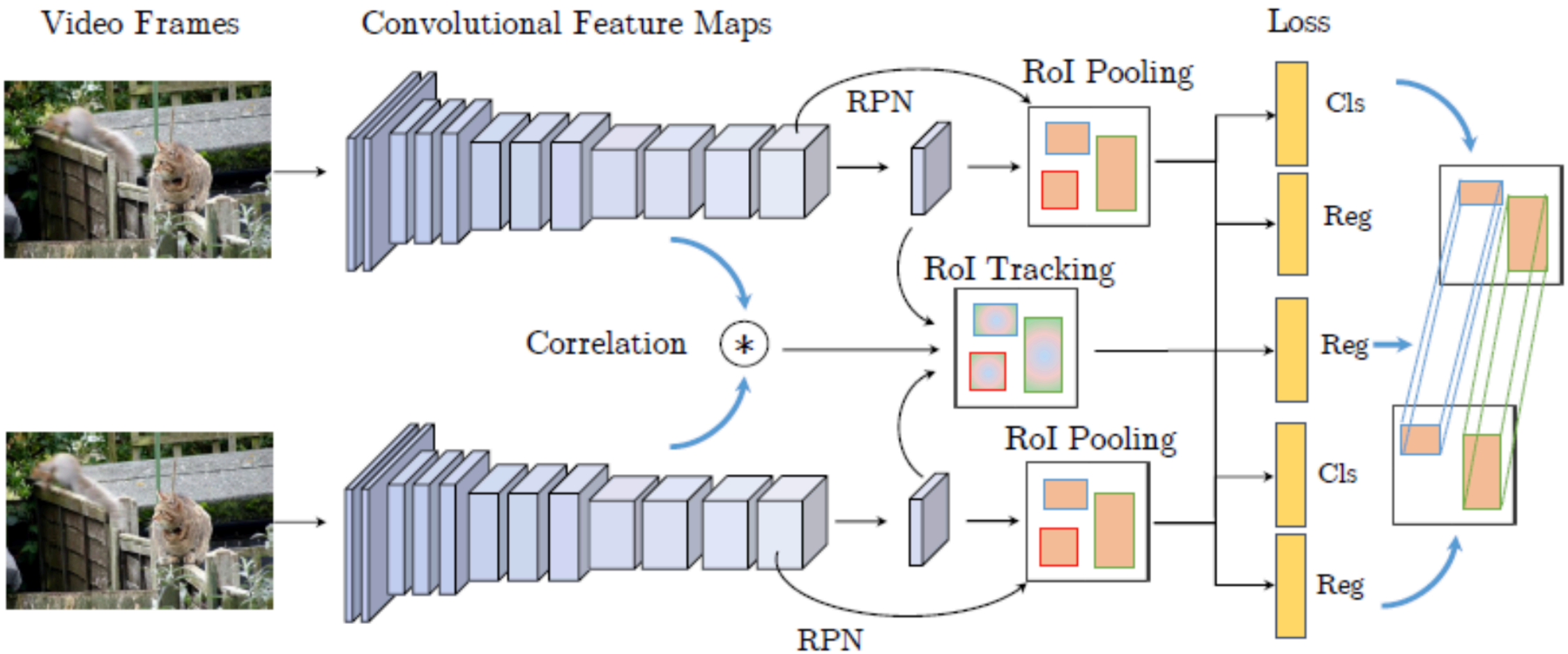
Slow Fast Networks

model	flow	pretrain	top-1	top-5	GFLOPs×views
I3D [3]		ImageNet	72.1	90.3	108 × N/A
Two-Stream I3D [3]	✓	ImageNet	75.7	92.0	216 × N/A
S3D-G [57]	✓	ImageNet	77.2	93.0	143 × N/A
Nonlocal R50 [52]		ImageNet	76.5	92.6	282 × 30
Nonlocal R101 [52]		ImageNet	77.7	93.3	359 × 30
R(2+1)D Flow [47]	✓	-	67.5	87.2	152 × 115
STC [7]		-	68.7	88.5	N/A × N/A
ARTNet [50]		-	69.2	88.3	23.5 × 250
S3D [57]		-	69.4	89.1	66.4 × N/A
ECO [59]		-	70.0	89.4	N/A × N/A
I3D [3]	✓	-	71.6	90.0	216 × N/A
R(2+1)D [47]		-	72.0	90.0	152 × 115
R(2+1)D [47]	✓	-	73.9	90.9	304 × 115
SlowFast 4×16, R50		-	75.6	92.1	36.1 × 30
SlowFast 8×8, R50		-	77.0	92.6	65.7 × 30
SlowFast 8×8, R101		-	77.9	93.2	106 × 30
SlowFast 16×8, R101		-	78.9	93.5	213 × 30
SlowFast 16×8, R101+NL		-	**79.8**	**93.9**	234 × 30

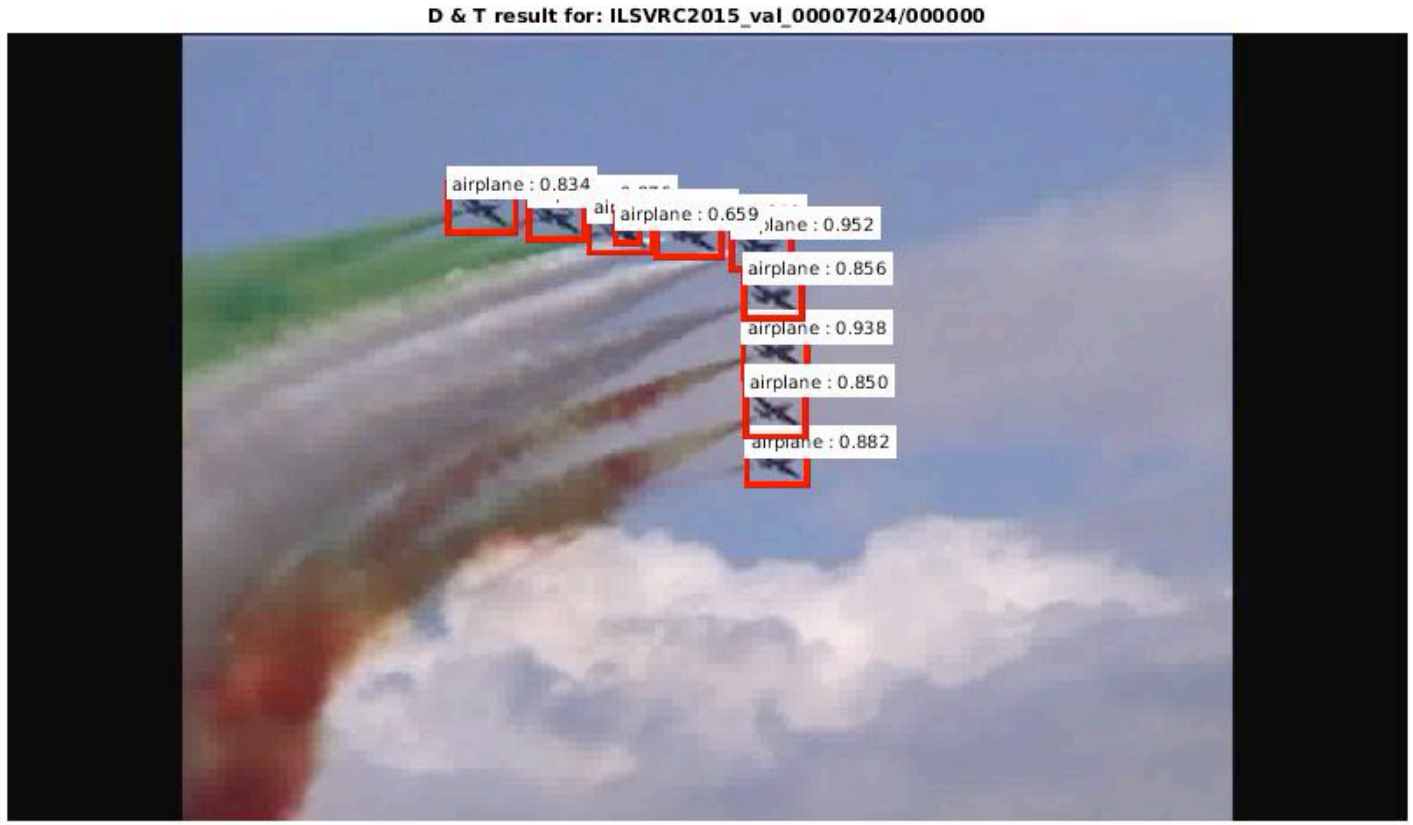
Tracking



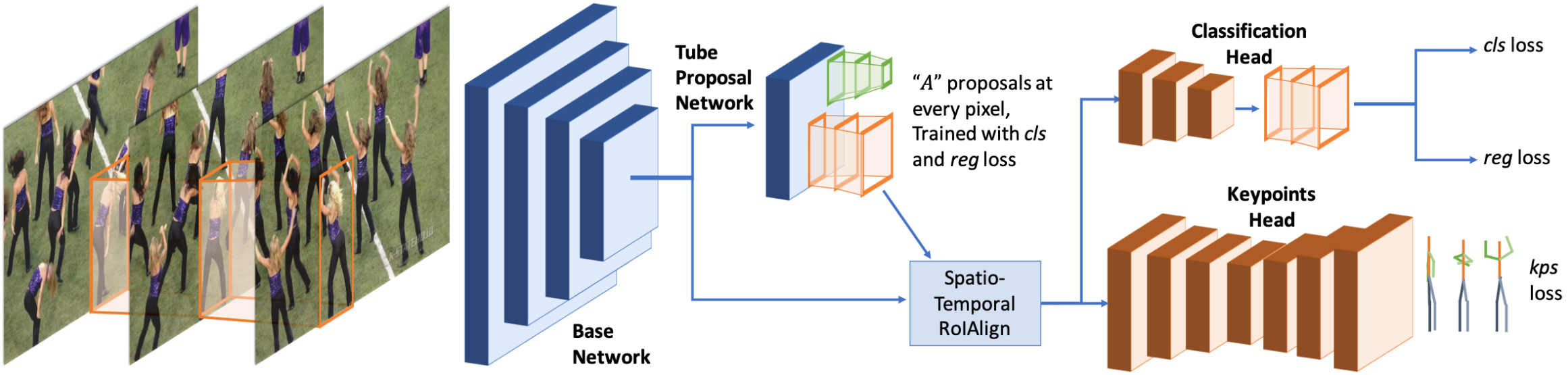




Feichtenhofer et al, Detect to Track and Track to Detect, ICCV 2017



Feichtenhofer et al, Detect to Track and Track to Detect, ICCV 2017



Girdhar et al, Detect and Track: Efficient Pose Estimation in Video, CVPR 2018

Let's Implement a Simple Tracker

Assume that at time t we have S_t detections, and at time t+1 we have S_{t+1} detections.

How can we link the detections in the two time steps, t & t+1?

Let's Implement a Simple Tracker

Assume that at time t we have S_t detections, and at time t+1 we have S_{t+1} detections.

How can we link the detections in the two time steps, t & t+1?

The i-th detection in S_t is linked to the j-th detection in S_{t+1}, if

$$max_{j \in S_{t+1}} \mathrm{sim}(i, j)$$

Let's Implement a Simple Tracker

Assume that at time t we have S_t detections, and at time t+1 we have S_{t+1} detections.

How can we link the detections in the two time steps, t & t+1?

The i-th detection in S_t is linked to the j-th detection in S_{t+1}, if

$$max_{j \in S_{t+1}} \mathrm{sim}(i, j)$$

$$\mathrm{sim}(i, j) = \mathrm{IOU}(\mathrm{box}_i, \mathrm{box}_j) \cdot \mathrm{score}_i \cdot \mathrm{score}_j$$